

SIMATS SCHOOL OF ENGINEERING

CAPSTONE PROJECT

BANK MANAGEMENT SYSTEM

CSA0488 – OPERATING SYSTEM

Faculty: Dr. Hemavathi

Group Members:

Akhilesh PK Shiva – 192210521

Barath S – 192211140

Dorwin S – 192211127

# Problem statements

- Develop an intuitive online banking platform accessible via web and mobile devices.
- Enable customers to easily navigate through their accounts, view transaction history, transfer funds, pay bills, and manage other banking services.
- Provide a streamlined process for users to apply for new accounts, loans, or credit cards online.
- Implement a secure payment gateway to facilitate online transactions, ensuring the protection of sensitive financial information such as credit card details and personal identification numbers (PINs).
- Comply with industry security standards like PCI DSS to safeguard customer data against unauthorized access or fraud.
- Integrate real-time account balance updates to provide customers with accurate information about their finances.
- Ensure seamless synchronization between online transactions and customers' account balances to prevent discrepancies.
- Enable customers to receive instant notifications for account activities, such as deposit confirmations, withdrawal alerts, or suspicious transaction alerts, via email, SMS, or push notifications.

# PROPOSED DESIGN

## IDENTIFYING THE KEY COMPONENTS:

- Allow users to input details such as account number, transaction type, amount, and any specific instructions or requests.
- Facilitate seamless transaction processing, including deposits, withdrawals, fund transfers, loan applications, and bill payments.
- Provide tools for analyzing account activity, generating statements, tracking spending patterns, and monitoring account balances.
- Implement features for customer support and assistance, such as helpdesk ticketing systems, live chat support, and knowledge bases.

## Functionality:

- Develop a user-friendly interface for easy navigation and interaction with banking services.
- Implement secure authentication mechanisms, encryption protocols, and fraud detection systems to ensure the safety of financial transactions.
- Provide convenient tools for users to view account details, update personal information, set up alerts, and manage account preferences.

## Architectural Design:

- Define the architecture of the bank management system, including frontend interfaces, backend databases, and middleware components.
- Implement robust security measures, including firewalls, intrusion detection systems, and data encryption, to protect sensitive customer information.
- Design the system to be scalable and reliable, capable of handling high volumes of transactions and concurrent user sessions without performance degradation.

# UI DESIGN

## Layout Design:

### 1. Homepage:

- The homepage serves as the main entry point for users, providing an overview of available banking services.
- Include sections for account login, new user registration, featured banking products, promotions, and quick links to commonly used banking features.

### 2. Account Dashboard:

- After logging in, users are directed to their personalized account dashboard.
- Display an overview of account balances, recent transactions, upcoming payments, and account alerts.
- Provide quick access to commonly used features such as fund transfers, bill payments, account statements, and customer support.

### 3. Transaction Pages:

- When initiating transactions such as fund transfers or bill payments, users are presented with dedicated pages for each transaction type.
- Include input fields for transaction details such as recipient information, amount, transfer date, and transaction notes.
- Display confirmation screens before finalizing transactions, allowing users to review and confirm details before proceeding.

### 4. Settings and Preferences:

- Allow users to access their account settings and preferences from the dashboard.
- Provide options to update personal information, change account settings, set up security features such as two-factor authentication, and manage communication preferences.

## **Feasible Elements Used:**

### **1. User Registration System:**

- Implement a user registration system with email verification to authenticate users and manage user accounts securely.

### **2. Real-Time Transaction Updates:**

- Integrate systems to provide real-time updates on account balances, transaction statuses, and account activities to users.

### **3. Customer Support Channels:**

- Offer customer support channels such as live chat support, email support, or a helpdesk system to assist users with inquiries and issues related to banking transactions.

## **Elements Positioning:**

### **1. Homepage Elements:**

- Place login and registration options prominently at the top of the homepage.
- Feature banking products, promotions, and quick links in easily accessible sections below the main header.

### **2. Account Dashboard Layout:**

- Organize account dashboard elements in a structured layout, with account summary sections at the top followed by quick access buttons and transaction history panels.

### **3. Transaction Pages:**

- Arrange transaction input fields and confirmation buttons in a clear and intuitive layout, guiding users through the transaction process step by step.

# CONCLUSION

The Bank Management System in Python using Tkinter is a software application designed to streamline and manage banking operations through a user-friendly graphical interface. Developed using the Python programming language and Tkinter library for the GUI, this system offers a range of features for efficient bank management. Users can perform essential banking operations such as creating, reading, updating, and deleting records, along with robust user authentication and authorization mechanisms.